Introduction to digital communication

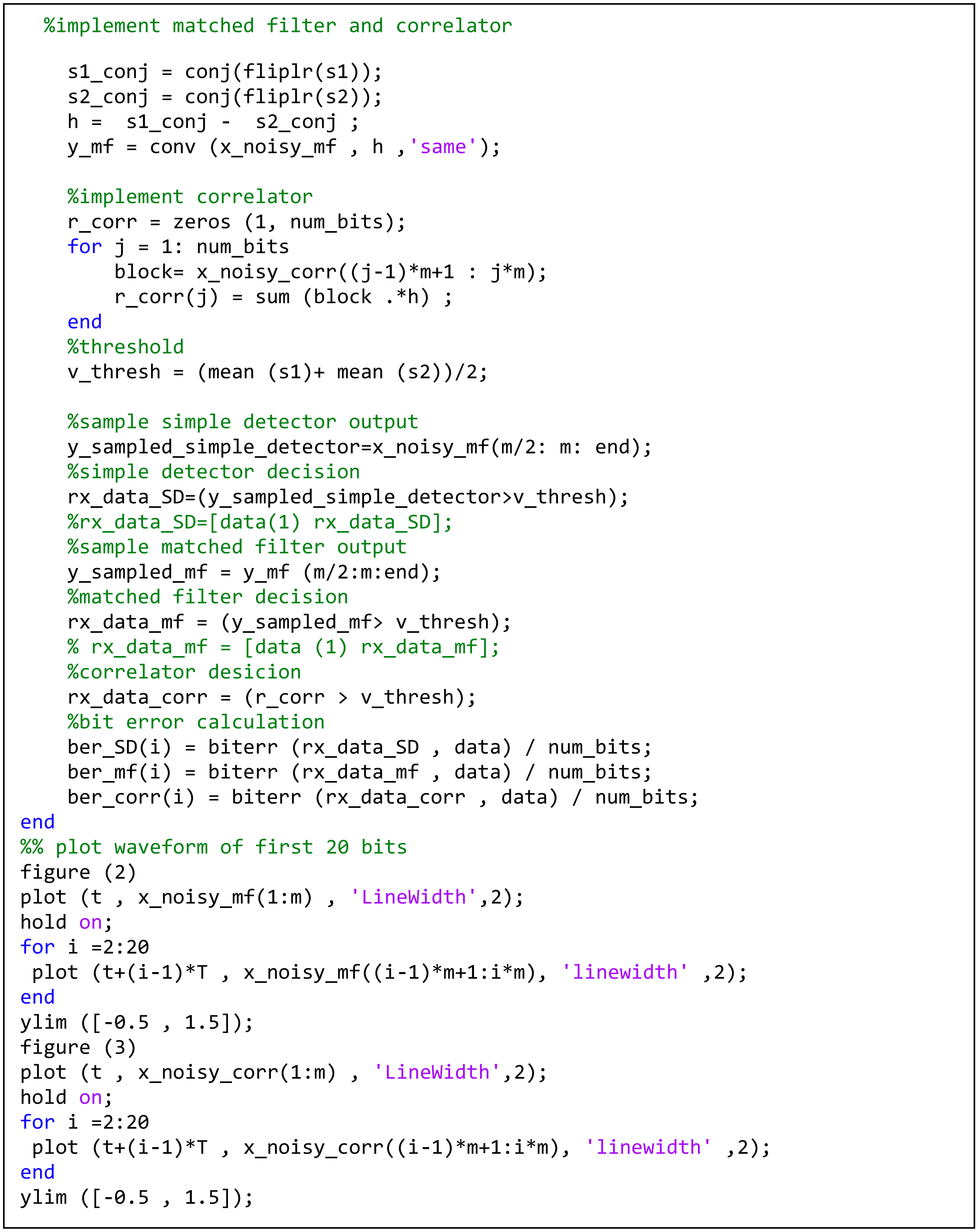
Final project

Part (1)

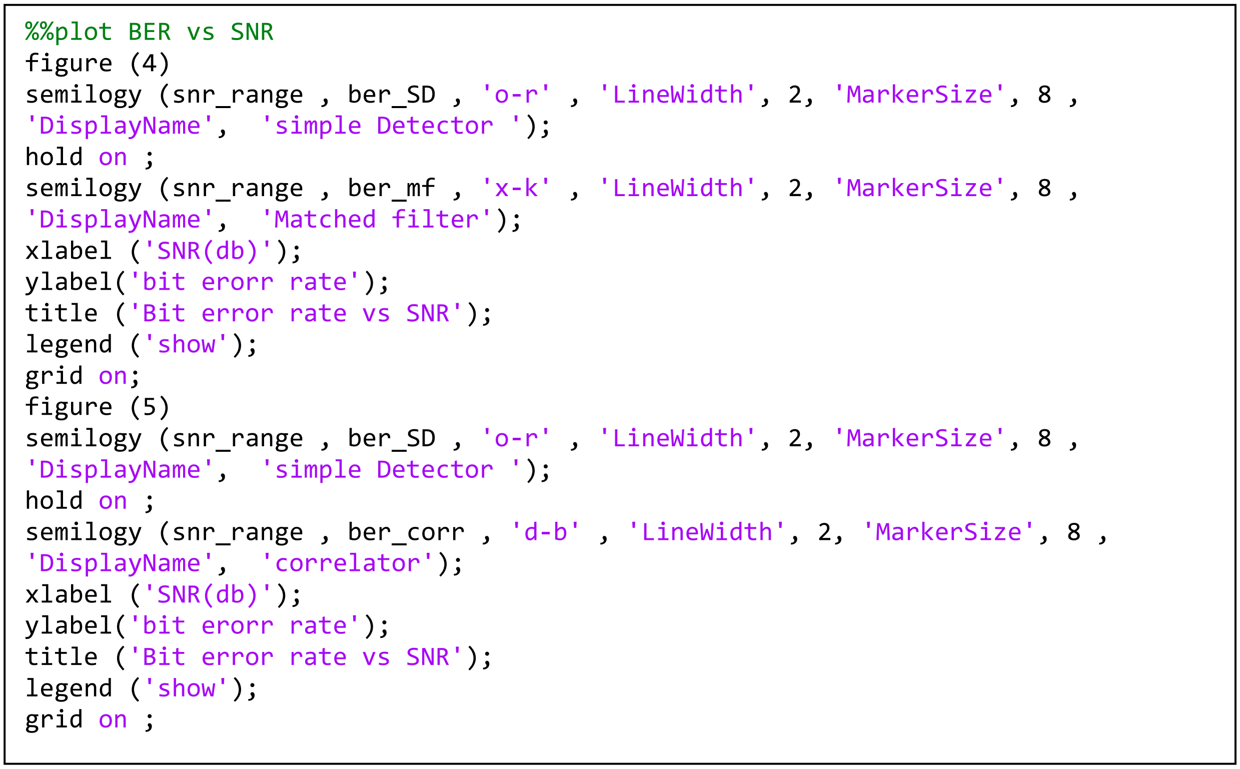
Performance of Matched filters and correlators

Name	ID
Asmaa Hassan Mokhtar Aboushady	19015430
Perihan Hossam Eldin Imam	20010392
Abdelraof Fathy Abdelraof Abdelrahman	20010777

```matlab
clear all
clc
close all
%%define system parameters
num_bits=1e5;
snr_range=0:2:30;
m = input ('enter the number of samples in the waveform: ');
T = 1;   %symbol period (normalized to 1)
n = m * num_bits ;  % Total number of samples
t = linspace (0,T,m);
sampling_instant=20;
%%define waveforms
% Generate s1(t) and s2(t)
expr1 = input ('enter the expression for s1: ','s');
s1 = eval(expr1);
expr2 = input ('enter the expression for s2: ','s');
s2 = eval(expr2);
% Generate random binary data
data = randi ([0,1], 1 , num_bits);
for i = 1:num_bits
    if data(i)==1
        x((i-1)*m+1:i*m) = s1;
    else
        x((i-1)*m+1:i*m) = s2;
    end
end
%%plot waveform of first 20 bits
figure (1)
plot (t, x(1:m), 'linewidth',2);
hold on ;
for i =2:20
    plot (t+(i-1)*T , x((i-1)*m+1:i*m), 'linewidth' ,2);
end
ylim ([-0.5 , 1.5]);
xlabel ('Time (s)');
ylabel('Voltage');
%%calculate transmitted signal power
signal_power = mean (abs(x).^2);
p = (1/n) * sum (abs(x).^2);
%%generate noise signal
noise_mf = zeros (1,n);
noise_corr = zeros (1,num_bits);
ber_mf = zeros(size(snr_range));
ber_corr= zeros(size(snr_range));
ber_SD= zeros (size (snr_range));
for i = 1:length (snr_range)
    snr = snr_range(i);
    noise_power= signal_power / (10^(snr/10));
    noise_std = sqrt(noise_power /2);
    noise_mf = noise_std* randn(1,n);
    noise_corr = noise_std* randn(1,n);
    %noise_mf = awgn (x, snr , 'measured');
    x_noisy_mf= x + noise_mf;
    x_noisy_corr= x+ noise_corr;
```



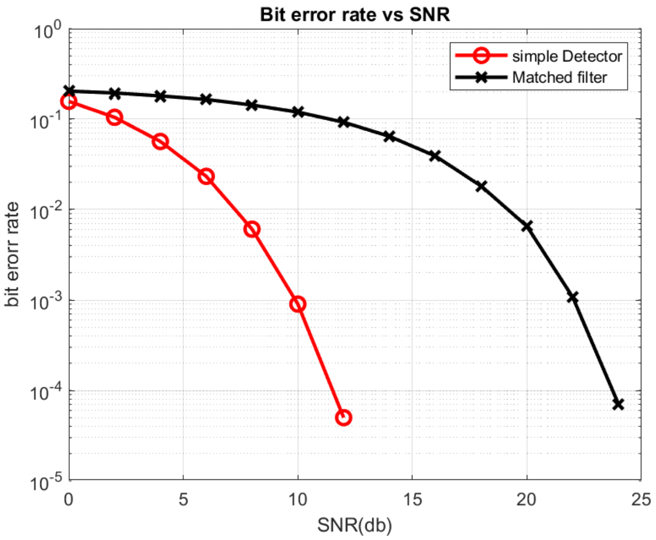
```
  %implement matched filter and correlator

    s1_conj = conj(fliplr(s1));
    s2_conj = conj(fliplr(s2));
    h =  s1_conj -  s2_conj ;
    y_mf = conv (x_noisy_mf , h ,'same');

    %implement correlator
    r_corr = zeros (1, num_bits);
    for j = 1: num_bits
        block= x_noisy_corr((j-1)*m+1 : j*m);
        r_corr(j) = sum (block .*h) ;
    end
    %threshold
    v_thresh = (mean (s1)+ mean (s2))/2;

    %sample simple detector output
    y_sampled_simple_detector=x_noisy_mf(m/2: m: end);
    %simple detector decision
    rx_data_SD=(y_sampled_simple_detector>v_thresh);
    %rx_data_SD=[data(1) rx_data_SD];
    %sample matched filter output
    y_sampled_mf = y_mf (m/2:m:end);
    %matched filter decision
    rx_data_mf = (y_sampled_mf> v_thresh);
    % rx_data_mf = [data (1) rx_data_mf];
    %correlator desicion
    rx_data_corr = (r_corr > v_thresh);
    %bit error calculation
    ber_SD(i) = biterr (rx_data_SD , data) / num_bits;
    ber_mf(i) = biterr (rx_data_mf , data) / num_bits;
    ber_corr(i) = biterr (rx_data_corr , data) / num_bits;
end
%% plot waveform of first 20 bits
figure (2)
plot (t , x_noisy_mf(1:m) , 'LineWidth',2);
hold on;
for i =2:20
 plot (t+(i-1)*T , x_noisy_mf((i-1)*m+1:i*m), 'linewidth' ,2);
end
ylim ([-0.5 , 1.5]);
figure (3)
plot (t , x_noisy_corr(1:m) , 'LineWidth',2);
hold on;
for i =2:20
 plot (t+(i-1)*T , x_noisy_corr((i-1)*m+1:i*m), 'linewidth' ,2);
end
ylim ([-0.5 , 1.5]);
```

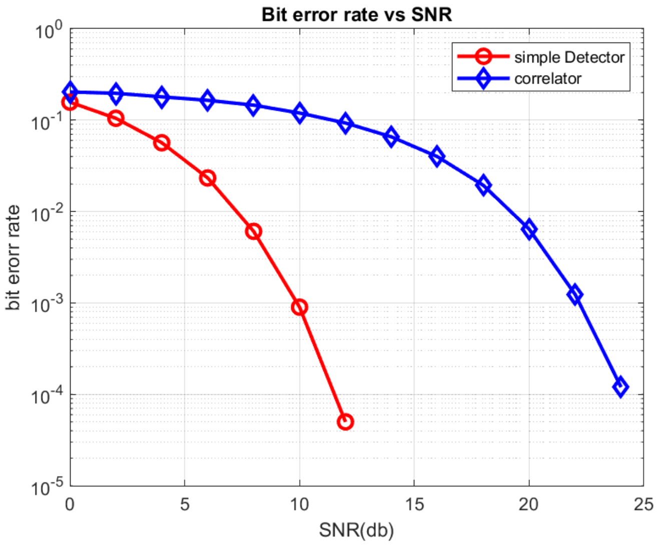


```
%%plot BER vs SNR
figure (4)
semilogy (snr_range , ber_SD , 'o-r' , 'LineWidth', 2, 'MarkerSize', 8 ,
'DisplayName',  'simple Detector ');
hold on ;
semilogy (snr_range , ber_mf , 'x-k' , 'LineWidth', 2, 'MarkerSize', 8 ,
'DisplayName',  'Matched filter');
xlabel ('SNR(db)');
ylabel('bit erorr rate');
title ('Bit error rate vs SNR');
legend ('show');
grid on;
figure (5)
semilogy (snr_range , ber_SD , 'o-r' , 'LineWidth', 2, 'MarkerSize', 8 ,
'DisplayName',  'simple Detector ');
hold on ;
semilogy (snr_range , ber_corr , 'd-b' , 'LineWidth', 2, 'MarkerSize', 8 ,
'DisplayName',  'correlator');
xlabel ('SNR(db)');
ylabel('bit erorr rate');
title ('Bit error rate vs SNR');
legend ('show');
grid on ;
```

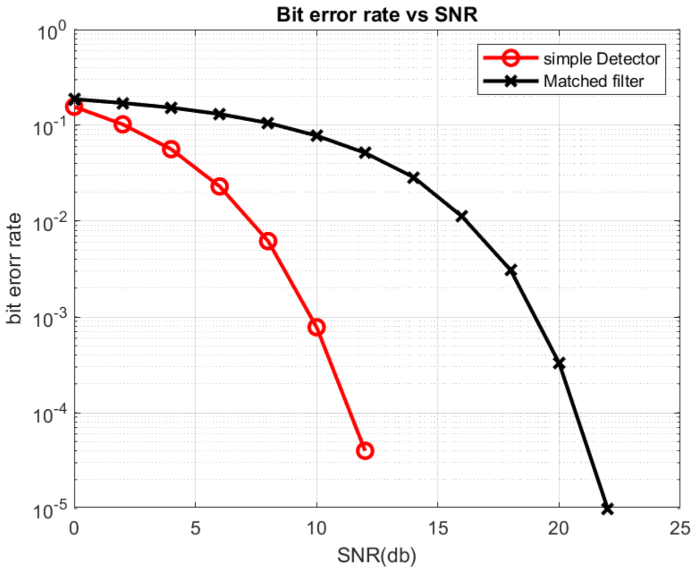
When the sampling instant =20

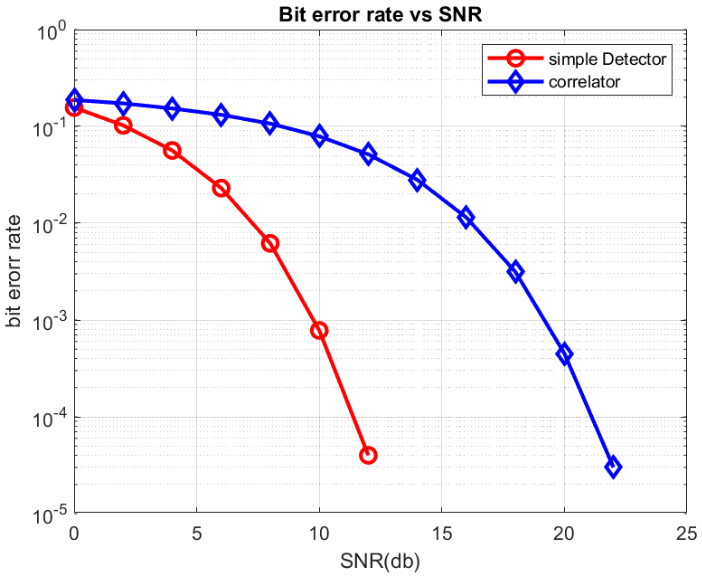
1) Num of samples =20



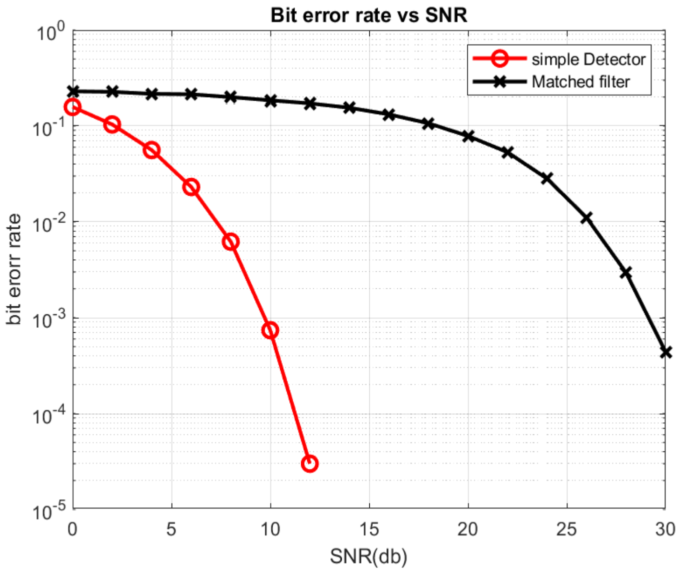


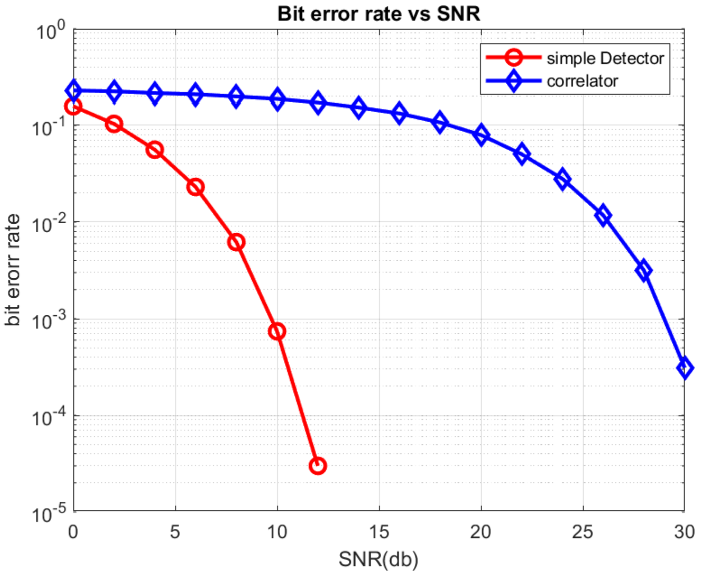
2) Num of samples = 10





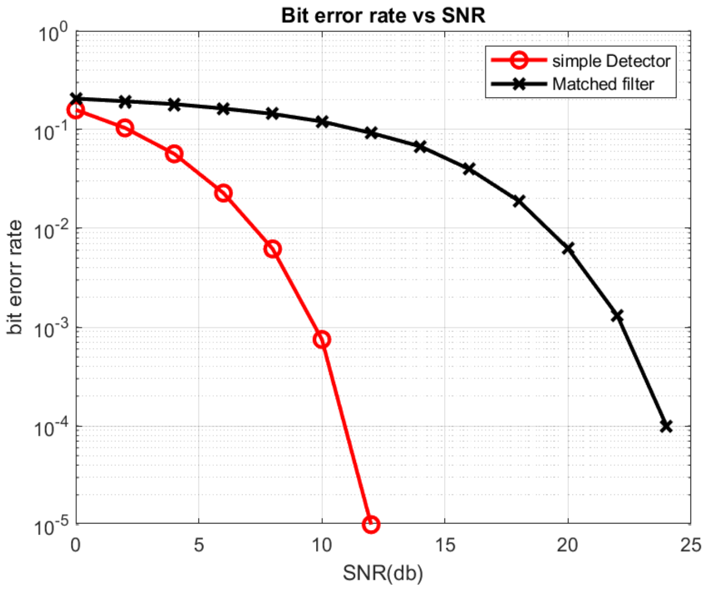
3) Num of samples = 100

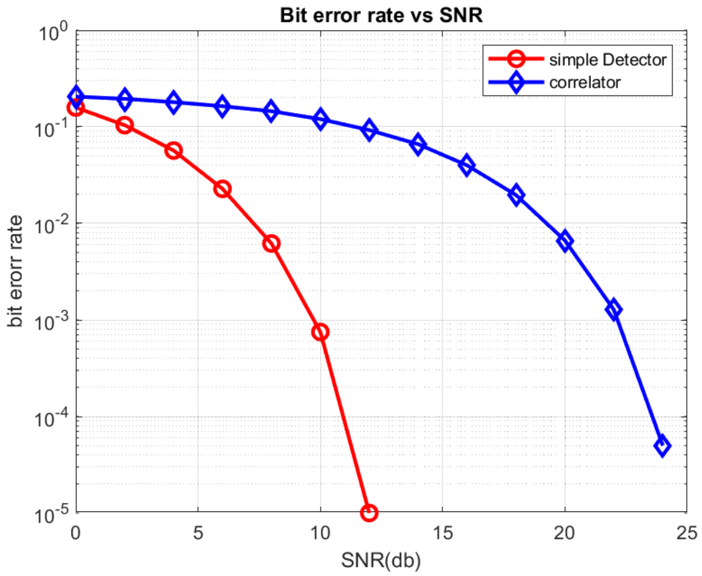




When the sample instant = m/2

Num of samples 20





Comment

the sampling instant only affects the matched filer detector and doesn’t affect neither simple detector nor the correlator.

Transmitted signal power

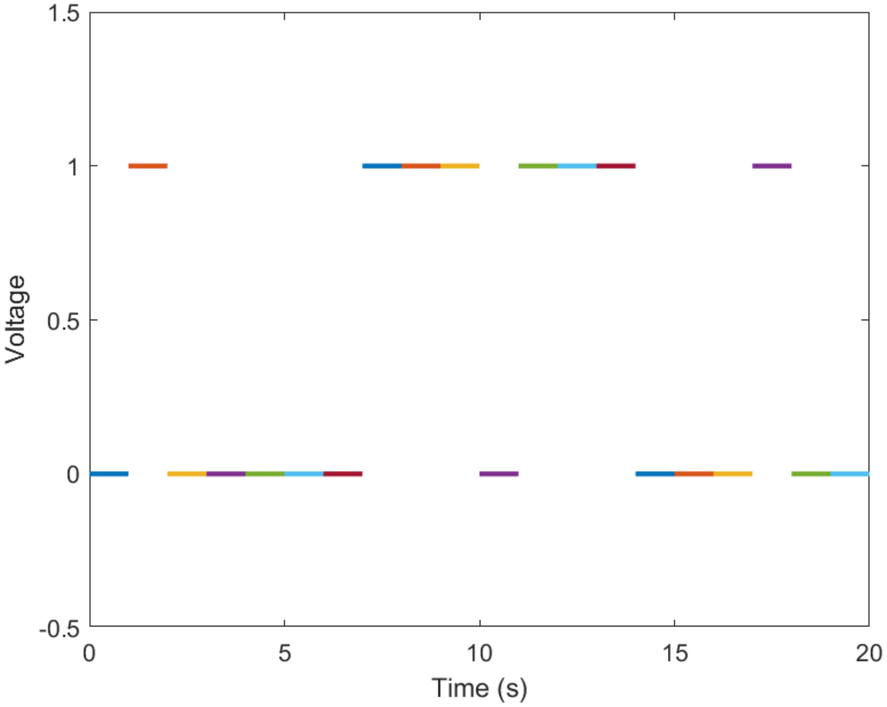
P(T)= mean(abs(x). ^2) or (1/n) * sum(abs(x). ^2) =0.5016

At which value of SNR, the system is nearly without error.

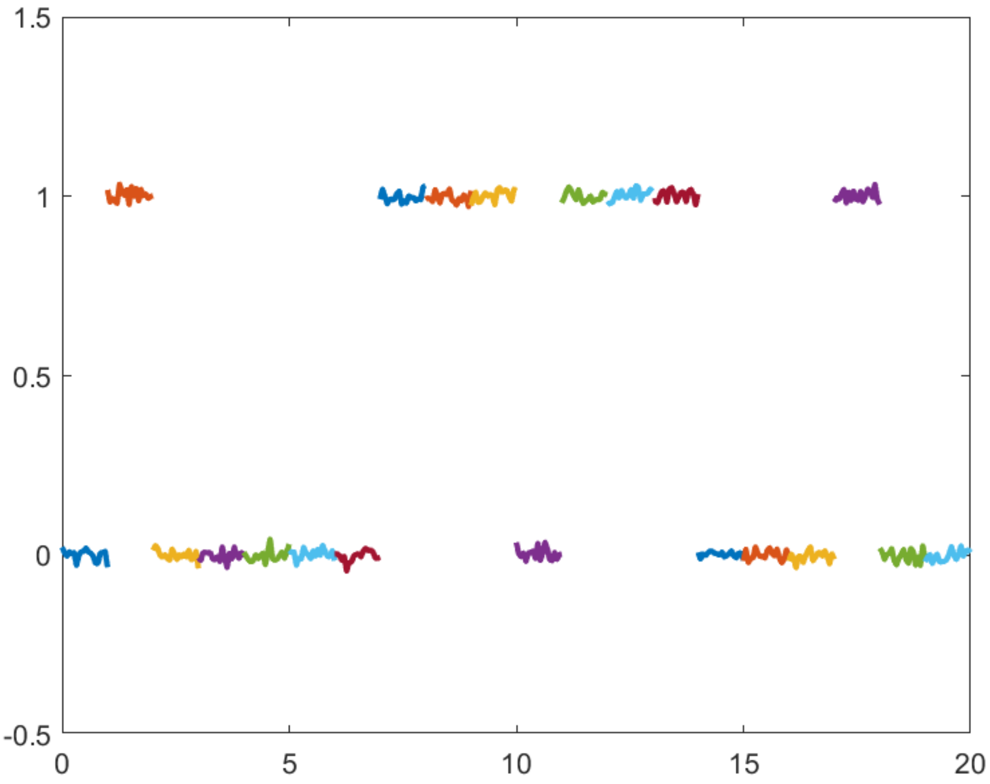
(for the given frame)?

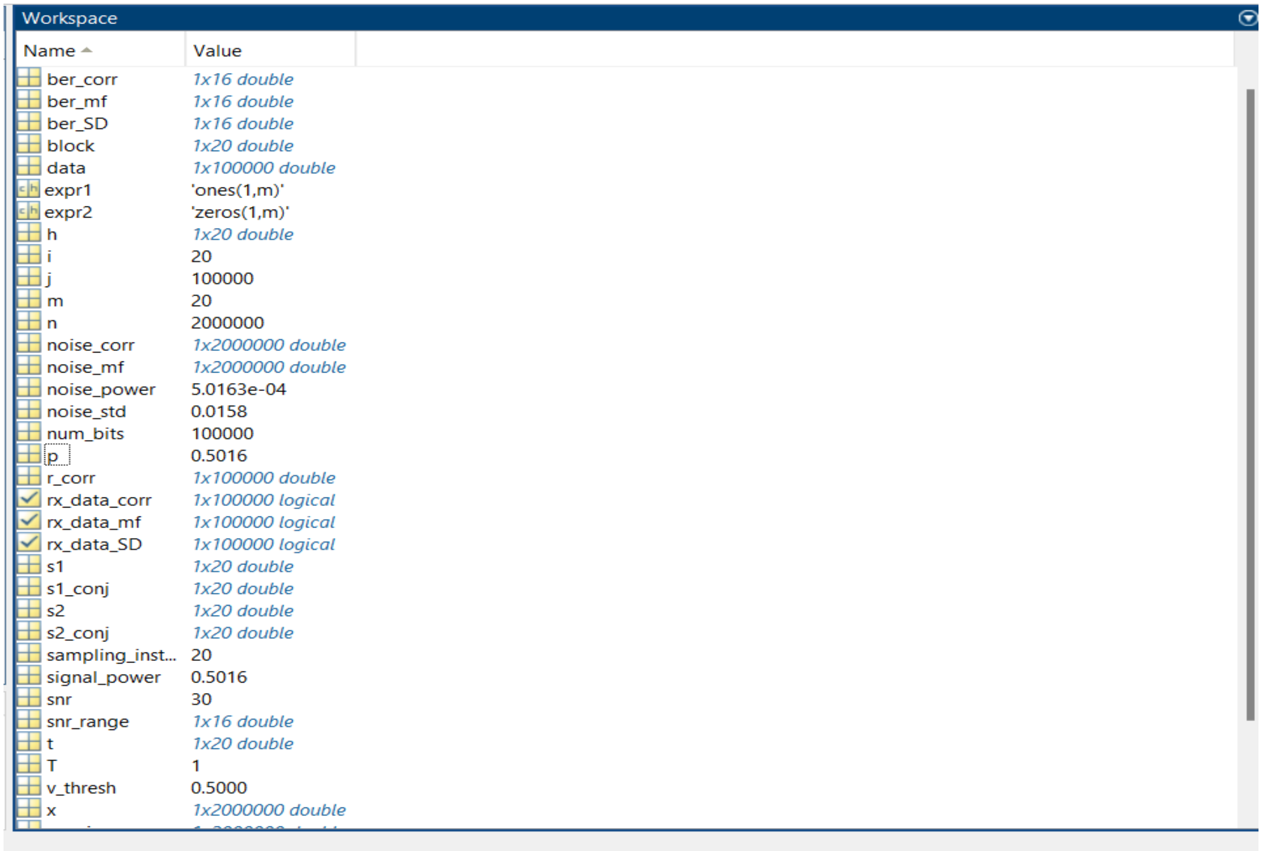
The higher the SNR the lower the BER for simple detector after 15dB there is no noise in case of MF and correlator there is no BER after 25dB.

the binary sequence after mapping to s1 and s2 without noise

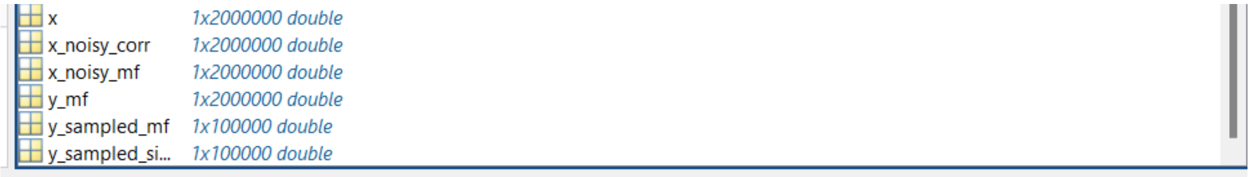


After adding noise





Name	Value
ber_corr	1x16 double
ber_mf	1x16 double
ber_SD	1x16 double
block	1x20 double
data	1x100000 double
expr1	'ones(1,m)'
expr2	'zeros(1,m)'
h	1x20 double
i	20
j	100000
m	20
n	2000000
noise_corr	1x2000000 double
noise_mf	1x2000000 double
noise_power	5.0163e-04
noise_std	0.0158
num_bits	100000
p	0.5016
r_corr	1x100000 double
rx_data_corr	1x100000 logical
rx_data_mf	1x100000 logical
rx_data_SD	1x100000 logical
s1	1x20 double
s1_conj	1x20 double
s2	1x20 double
s2_conj	1x20 double
sampling_inst...	20
signal_power	0.5016
snr	30
snr_range	1x16 double
t	1x20 double
T	1
v_thresh	0.5000
x	1x2000000 double



x	1x2000000 double
x_noisy_corr	1x2000000 double
x_noisy_mf	1x2000000 double
y_mf	1x2000000 double
y_sampled_mf	1x100000 double
y_sampled_si...	1x100000 double